

REMARKS

This is a response to the final Office Action mailed June 14, 2007.

Claims 1, 3-5, 7-11, 14, 19-21, 23, 27, 28, 33, 37, 39, 45-48, 52, 54-58, 60-62, 64, 65 and 67-70 are amended, and claims 73-80 are new. Example support for the claims can be found as follows: claim 1 (p.7, line 15-p.8, line 2; p.5, lines 9-20, and former claim 4), claims 5 and 9 (p.32, line 26-p.33, line 26), claim 7 (p.6, lines 4-8), claims 8, 20 and 27 (Fig. 8), claim 19 (p.17, lines 12-18), claim 28 (p.3, lines 2-4; p.12, line 27, Fig. 8), claim 45 (p.8, lines 18-26), claim 48 (p.7, line 20), claims 74, 75 (p.16, lines 1-8), claim 76 (p.25, lines 7-9, p.24, lines 11-14), claim 77 (p.8, lines 18-21), claim 78 (p.7, line 15 to p.9, line 2). No new matter is added.

Claim 11 has been amended in response to the rejection under 35 U.S.C. §112, second paragraph.

Claims 1, 3, 7-10, 28-36, 41-45, 47 and 52-72 have been rejected under 35 U.S.C. 102(b) as being anticipated by Wu et al. (US 5,987,256) (Wu). Applicants also address Davis (US 6,643,696) which was cited against claim 4, since claim 1 includes subject matter from former claim 4. Applicants respectfully traverse this and the other rejections.

Wu uses a precompiler which translates standard HTML, Java or other programs to display-oriented language codes for use by the thin client platform (col. 2, lines 56-61, col. 3, lines 6-12).

Davis, applied to former claim 4, provides a process for tracking client interaction with a resource such as a web page downloaded from a server. In particular, an HTML document is downloaded from a server, then images specified by first embedded URLs are downloaded. A second embedded URL in the document points to a first executable program that runs on a server. A third embedded URL in the document points to a second executable program that is downloaded to and runs on the client. When the images specified by the first embedded URLs are downloaded, the first executable program on the server runs. The server can capture identifying information from the client. The second executable program determines the current time when it initializes, and the current time when the user leaves the HTML document. The elapsed time is then reported to the server (col. 5, line 40-col. 6, line 16).

Claim 1 is patentable over the cited references because they do not disclose or suggest a markup language description, where one or more source files define a connection to an external data source for data, and where the data is for rendering on a user interface. The Office Action, p.14,

acknowledges that Wu is not concerned with one or more source files that define a connection to an external data source for data. However, Davis's use of embedded URLs that point to executable programs that run on a server or a client do not cure this deficiency, at least because the embedded URLs do not involve data which is for rendering on a user interface. Instead, the embedded URLs are merely pointers that are not rendered. Further, Davis does not access data and compile the data, either alone or as part of compiling a markup language description, in response to a request which is received for particular content. Instead, the executable programs are stored and made available on servers (Davis, col. 17, lines 32-47). Claim 1 is therefore clearly allowable.

Claim 7 includes a limitation from former claim 28 regarding a plug-in. The Office Action, p.5, asserts that Wu provides a plug-in at col. 17, lines 10-22. However, the cited passage refers to a display routine of the thin client platform which is executed. The process executes applets which are included in a file. There is no mention of a rendering entity which is a plug-in to a browser as claimed. Further, the thin client platform of Wu has limited processing resources and is therefore incompatible with a standard HTML rendering program which could accommodate a plug-in.

Further the Office Action, p.29 (par. 89) asserts that a plug-in is "any type of computer program that interacts with a main host." Applicants respectfully submit that this interpretation is in conflict with MPEP 2111.01 (Plain Meaning), which states that terms of a claim must be given the meaning that they would have to a person of ordinary skill in the art in question at the time of the invention. For example, the enclosed definition from the Microsoft Computer Dictionary (2002) states that a plug-in is "a small software program that plugs into a larger application to provide added functionality." Applicants' specification also provides an example of a Flash Player embedded in a web client as a plug-in (p.6, lines 4 and 5). Thus, the use of a rendering entity as a plug in to a browser, as set forth in claim 7, provides the important advantage of adding additional functionality to the browser. This is an important distinction that is not disclosed or suggested by the cited references. Claim 7 is therefore clearly allowable.

Claim 8 is patentable over Wu at least because Wu does not provide a reference to transformed media content in executable code. Regarding col. 2, line 47-50 of Wu, this only indicates that a first data set is translated to a second data set in an intermediate language which is adapted for execution by a target device, and the second data set is sent to the target device for execution. However, Wu has no need to provide a reference to media content since the second data

set provides all rendering instructions in the form of simplified graphics primitives that define rectangles, text and bitmaps (col. 5, lines 5-7 and col. 6, lines 12-15). Further, Wu does not provide media content to the target device, where the media content is not compiled. Instead, all of Wu's rendering instructions are compiled (col. 4, lines 49-65). Claim 8 is therefore clearly allowable.

Claim 9 sets forth that a step of compiling includes converting a mark-up language description to ActionScript, and compiling the ActionScript into ActionScript byte code. The claim clarifies that it involves "ActionScript". The Examiner is requested to interpret the term *t* according to the meaning it would have to a person of ordinary skill in the art in question at the time of the invention. The portion of Wu which is cited as disclosing this feature is col. 17, lines 47-50, which refers to translating Java byte code to a reduced byte code. However, converting Java byte code to a reduced byte code does not involve converting a mark-up language description to ActionScript, and compiling the ActionScript into ActionScript byte code. Claim 9 is therefore clearly allowable.

Independent claim 28 refers, in part, to compiling a mark-up language description of particular content to create executable code for a plug-in to a browser, where the mark-up language description references a media file comprising at least one of audio, video and a movie, and transmitting the executable code and the media file from a server to the plug-in, where the media file is not compiled. In contrast, the thin client platform of Wu cannot handle a media file as claimed, but only processes HTML and Java code (col. 4, lines 25-37) to provide all rendering instructions in the form of simplified graphics primitives (col. 5, lines 5-7 and col. 6, lines 12-15).

Regarding the data sets referred to by Wu at col. 4, lines 43-48, this merely refers to the fact that the compiled code can be stored for subsequent use on a storage medium at a VCD (compact video disk), DVD (digital video disk) or the world wide web. There is no disclosure or suggestion of providing a markup language description with a reference to a media file as claimed.

Claim 28 and its dependent claims are therefore clearly allowable.

Claims 33 and 41 are patentable at least for the reasons discussed above in connection with claim 7.

Claim 45 is patentable at least for the reasons discussed in connection with claim 1.

Claim 52 sets forth that a request for content includes an indication that identifies a type of a rendering entity, and that executable code is created specific for the type of rendering entity. Regarding col. 17, lines 10-22 of Wu, this relates to a display process of the thin client engine and

does not involve a request for content which is received at a server. The Examiner is requested to clarify this rejection.

The remaining dependent claims are similarly patentable. For example, claim 58 refers to an element which is identified by a markup language tag, and which refers to a media file comprising at least one of audio, video and a movie. This claim is patentable for the reasons discussed above in connection with claim 28. Claim 58 is therefore clearly patentable.

Claim 62 refers to an element which references a .SWF file. The Office Action states that a .SWF file is deemed an example of a graphic object. Applicants respectfully submit that this definition is in conflict with the plain meaning of the claim terms. Further, Wu states at col. 5, line 18 to col. 6, line 7, that HTML rendering involves fixing the coordinates of all graphic objects specified by the HTML code, such as by word-wrapping paragraphs and choosing colors. However, such HTML rendering does not involve a reference to a .SWF file, and it is clear that Wu's thin client platform can only render simplified graphics primitives, as discussed. Claim 62 is therefore clearly patentable.

New claim 73 sets forth that the rendering entity is a Flash player. Again, the thin client platform of Wu does not provide the capability of a Flash player. Claim 73 is therefore clearly patentable.

Claim 6 has been rejected under 35 U.S.C. 103(a) as being unpatentable over Wu.

This claim is patentable at least by virtue of its dependence on an independent claim which is patentable as discussed herein.

Claim 11 has been rejected under 35 U.S.C. 103(a) as being unpatentable over Wu in view of Russell (US2002/0069420).

This claim is patentable at least by virtue of its dependence on an independent claim which is patentable as discussed herein. Further, claim 11 sets forth that compiling (of a mark-up language description of particular content to create executable code) and transmitting (of the executable code) are only performed if authenticating is successful, where different types of authenticating are provided for different types of content and/or for each item of content. Russell is concerned with delivering content such as movie files over a network. However, there is no disclosure or suggestion of allowing compiling based on authenticating as claimed. Further, a person of ordinary skill in the art would not be led to combine these references as suggested because the thin client of Wu can only

handle simplified graphics primitives, but cannot handle movie files, music files, or video games files with which Russell is concerned (par. 4).

The Office Action, p.29, asserts that Wu is not limited to handling simplified graphics primitives because Wu refers to running HTML and Java programs on a TV set top box, handheld device, or digital video disk player (col. 2, lines 59-63). However, Wu's rendering engine 12 (Fig. 1) is a separate component which operates in a separate surf_HTML mode (col. 17, lines 3-6). There is no indication that the surf_HTML mode provides video and audio data, for instance, to the device. It can only be concluded that the TV set top box or digital video disk player, for instance, receive video and audio data from a conventional television signal or a DVD, which do not use a markup language. Claim 11 is therefore clearly patentable.

Claims 4, 5, 13, 21-27, 37-40 and 46 have been rejected under 35 U.S.C. 103(a) as being unpatentable over Wu in view of Davis et al. (US6,643,696) (Davis).

Davis provides a process for tracking client interaction with a resource such as a web page downloaded from a server, as discussed above in connection with claim 1. It would not be obvious to combine Wu and Davis because, as discussed, regarding Davis's use of embedded URLs that point to executable programs that run on a server or a client, the embedded URLs do not involve data which is for rendering on a user interface. Instead, the embedded URLs are merely pointers that are not rendered. Further, Davis does not access and compile data as claimed.

Independent claim 21, for instance, clearly distinguishes over Wu and Davis at least because it sets forth receiving a request for content that includes data other than code, where the data is for rendering on a user interface at a client, and accessing a mark-up language description associated with the content, where the mark-up language description defines a connection to an external data source for the data. Neither Wu nor Davis, taken alone or in combination, disclose or suggest these features. Claim 21 and its dependent claims are therefore clearly patentable.

Claim 27 is patentable for the reasons discussed above in connection with claim 8.

Claim 37 is patentable at least for the reasons discussed above in connection with claim 21.

Claims 14-20 and 48-51 have been rejected under 35 U.S.C. 103(a) as being unpatentable over Wu in view of Wagner (US6,085,224).

Independent claim 14 sets forth, in part, accessing first code associated with particular content, where the first code includes a mark-up language description and a scripting language

description, and compiling the mark-up language description and scripting language description to create combined executable code.

Wagner is concerned with detecting hidden data such as cookies in a data stream. Wagner states at col. 15, line 61 to col. 16, line 15 that embedded commands can be used to activate or execute a program or applet. Programs which may be activated by an embedded command include Java script programs. For example, an HTML tag can be used to invoke a Java applet. The Examiner assets that it would be obvious to modify Wu by including scripting language embedded within a markup language file. However, Wu teaches against the proposed combination since Wu sets forth that a translation process is selected according to the identified object specifying language (e.g., HTML or Java) (col. 2, lines 31-37). Wu further provides separate precompilers for HTML and Java, respectively (Col. 4, lines 66-67, Figs. 3 and 4). Thus, the translation/compiling process is clearly specific to the object specifying language. Accordingly, the proposed combination could only be made impermissibly in view of Applicants' invention.

Accordingly, claim 14 and its dependent claims are clearly patentable.

Claim 19 sets forth that the markup language description includes elements which are identified by markup language tags, and at least one of the elements provides a script source of a scripting language description. For example, see the specification, p.17, lines 12-19. Neither Wu nor Wagner, taken alone or in combination, disclose or suggest this feature.

Dependent claim 20 is patentable for the reasons discussion in connection with claim 8.

Claim 48 sets forth, in part, that script code of a scripting language description is contained within script tags of a markup language description. See the specification, p.7, line 20. Neither Wu nor Wagner, taken alone or in combination, disclose or suggest this feature.

Conclusion

In view of the above, each of the pending claims is believed to be in condition for immediate allowance. The Examiner is therefore respectfully requested to pass this application on to an early issue.

The Examiner's prompt attention to this matter is greatly appreciated. Should further questions remain, the Examiner is invited to contact the undersigned attorney by telephone.

The Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 501826 for any matter in connection with this response, including any fee for extension of time, which may be required.

Respectfully submitted,

Date: October 15, 2007 By: /Ralph F. Hoppin/
Ralph F. Hoppin
Reg. No. 38,494

VIERRA MAGEN MARCUS & DENIRO LLP
575 Market Street, Suite 2500
San Francisco, California 94105-4206
Telephone: (415) 369-9660, x214
Facsimile: (415) 369-9665